10/15/2020

MVA ASSIGNMENT 5

MEMBER INFORMATION

Ruixin Yang (RUID: 197000459)

Aishwarya Senthilvel (RUID: 199001269)

CLUSTERING ANALYSIS

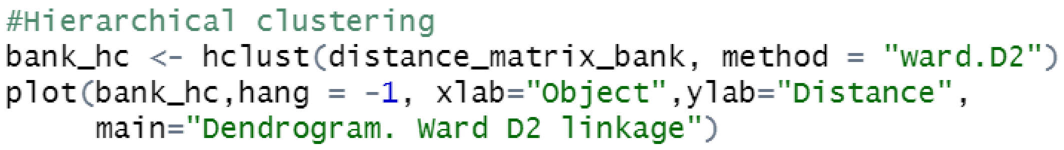
We perform Cluster analysis to group a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other clusters.

Cluster analysis is a powerful data-mining tool for any organization especially in that of our case: A Bank Data, that needs to identify discrete groups of customers, sales transactions, and other types of behaviors.

TWO-STEP CLUSTERING

We have performed two-step clustering that handles scaled down bank data and ordinal data in the same model, and it automatically selects the number of clusters.

HIERARCHICAL CLUSTERING



```
#Hierarchical clustering
bank_hc <- hclust(distance_matrix_bank, method = "ward.D2")
plot(bank_hc,hang = -1, xlab="Object",ylab="Distance",
     main="Dendrogram. Ward D2 linkage")
```

The hierarchical cluster analysis follows three basic steps:

1) calculate the Euclidean distances

2) linkage of clusters

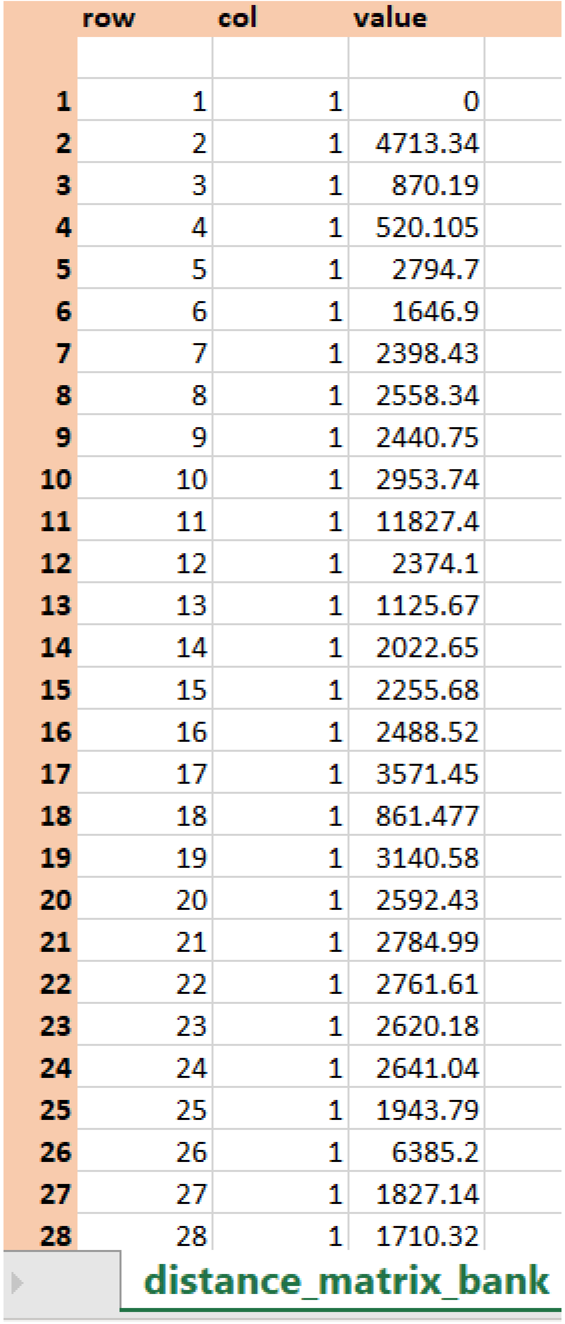
3) choosing a solution by selecting the right number of clusters

CALCULATING EUCLIDEAN DISTANCES

```
# Calculating Euclidean distance matrix for bank data
distance_matrix_bank<-dist(bank[1:30,],method="euclidean")
```

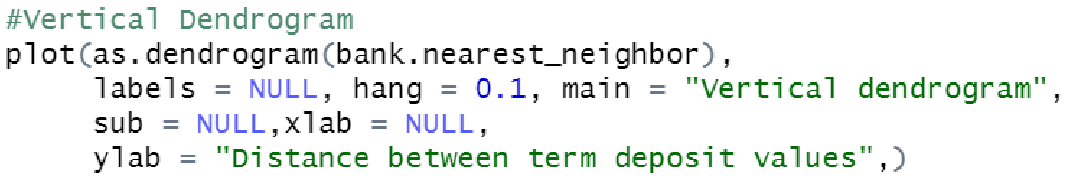
DISTANCE MATRIX

We have considered the first 30 customers as our sample with respect to the large data size.



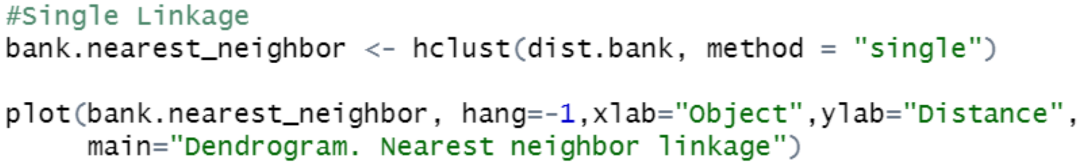
	row	col	value
1	1	1	0
2	2	1	4713.34
3	3	1	870.19
4	4	1	520.105
5	5	1	2794.7
6	6	1	1646.9
7	7	1	2398.43
8	8	1	2558.34
9	9	1	2440.75
10	10	1	2953.74
11	11	1	11827.4
12	12	1	2374.1
13	13	1	1125.67
14	14	1	2022.65
15	15	1	2255.68
16	16	1	2488.52
17	17	1	3571.45
18	18	1	861.477
19	19	1	3140.58
20	20	1	2592.43
21	21	1	2784.99
22	22	1	2761.61
23	23	1	2620.18
24	24	1	2641.04
25	25	1	1943.79
26	26	1	6385.2
27	27	1	1827.14
28	28	1	1710.32

PLOTTING VERTICAL DENDROGRAM

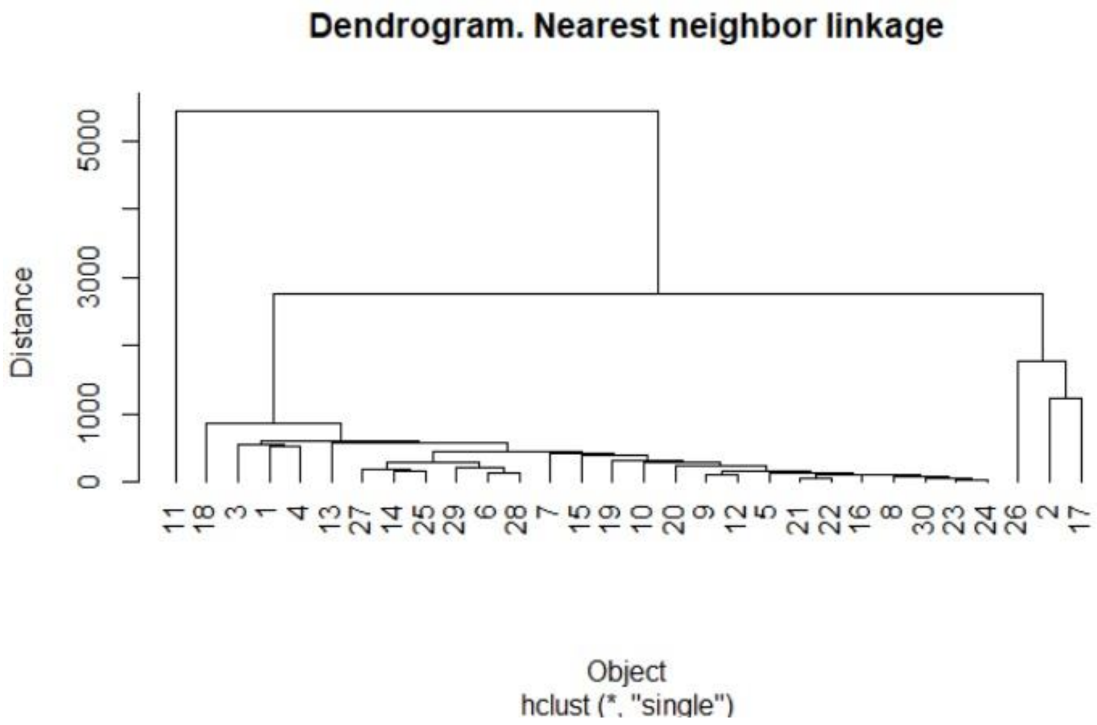


```
#Vertical Dendrogram
plot(as.dendrogram(bank.nearest_neighbor),
     labels = NULL, hang = 0.1, main = "Vertical dendrogram",
     sub = NULL,xlab = NULL,
     ylab = "Distance between term deposit values",)
```

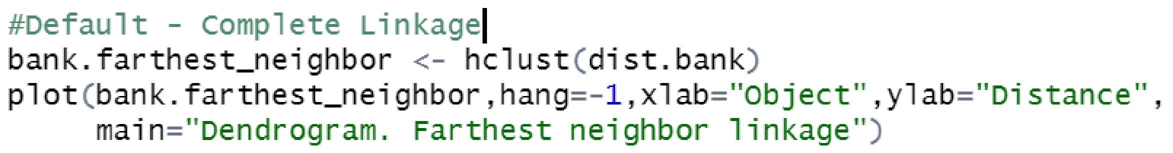
SINGLE LINKAGE



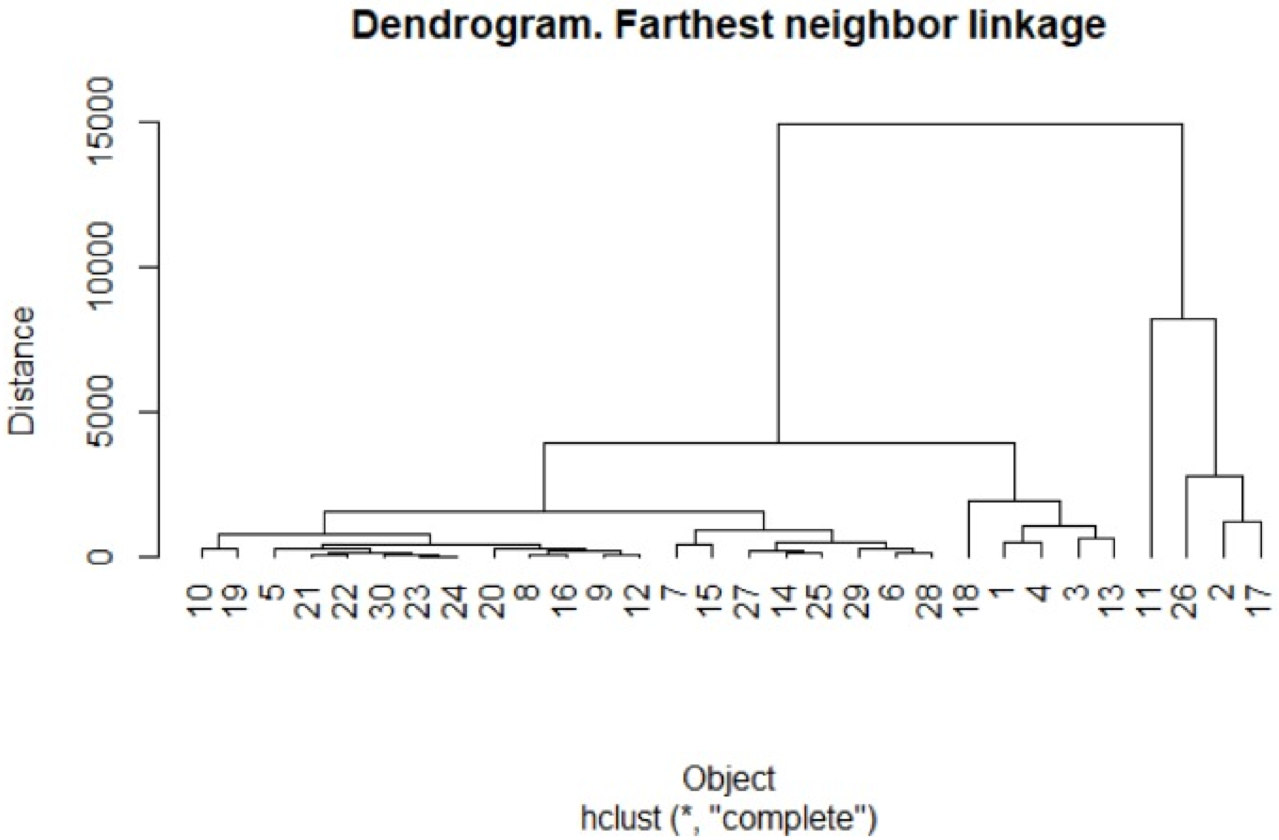
```
#Single Linkage
bank.nearest_neighbor <- hclust(dist.bank, method = "single")

plot(bank.nearest_neighbor, hang=-1,xlab="Object",ylab="Distance",
     main="Dendrogram. Nearest neighbor linkage")
```



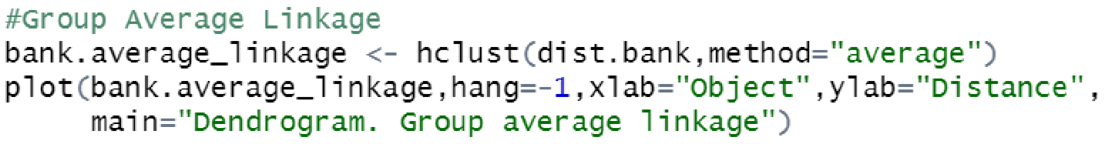
COMPLETE LINKAGE



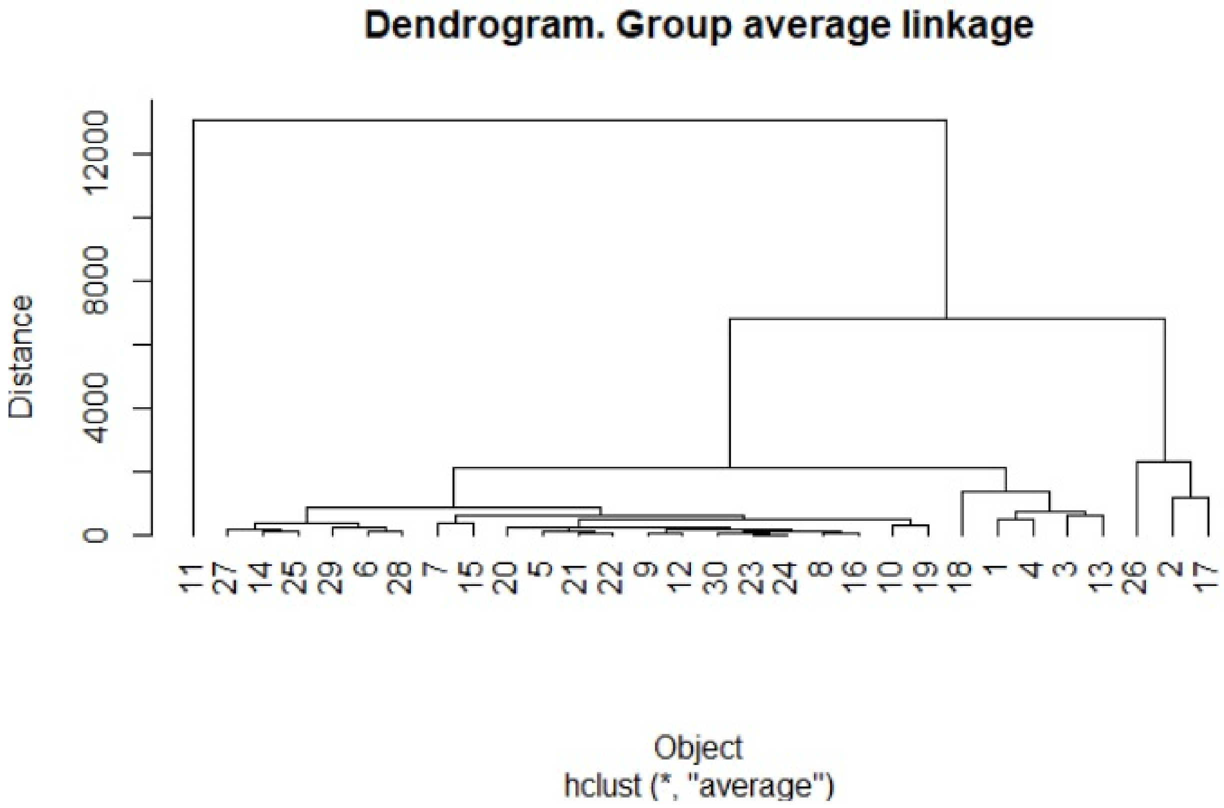
```
#Default - Complete Linkage
bank.farthest_neighbor <- hclust(dist.bank)
plot(bank.farthest_neighbor,hang=-1,xlab="Object",ylab="Distance",
     main="Dendrogram. Farthest neighbor linkage")
```



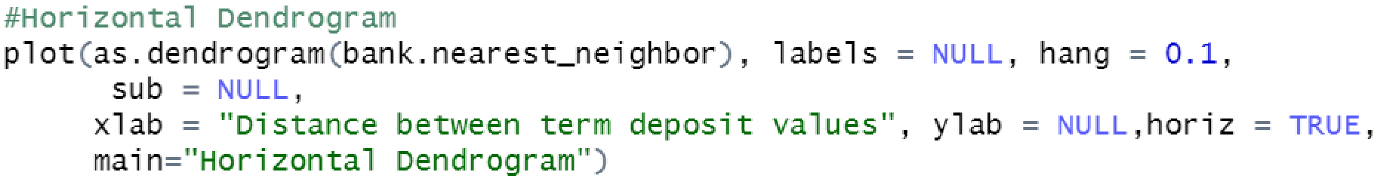
AVERAGE LINKAGE



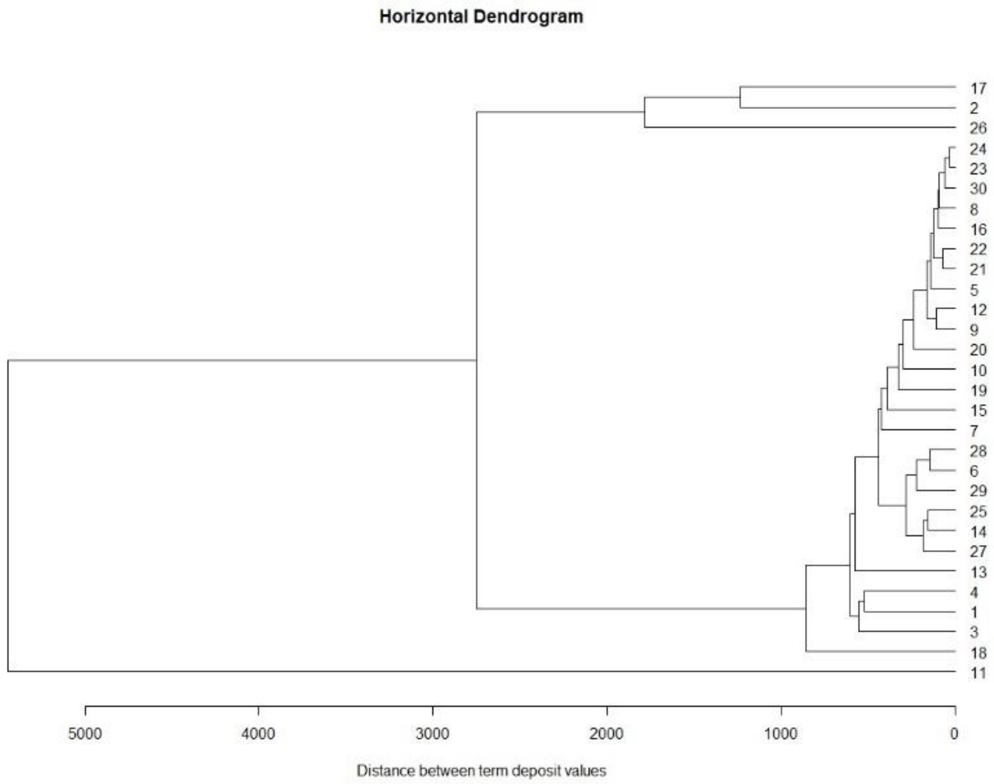
```
#Group Average Linkage
bank.average_linkage <- hclust(dist.bank,method="average")
plot(bank.average_linkage,hang=-1,xlab="Object",ylab="Distance",
     main="Dendrogram. Group average linkage")
```



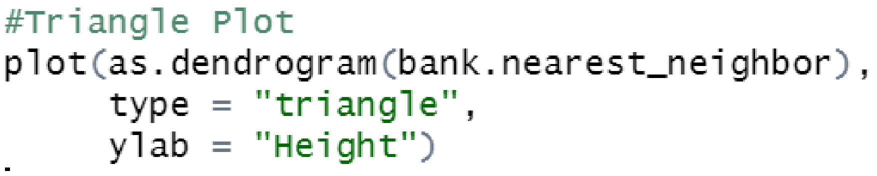
PLOTTING HORIZONTAL DENDROGRAM



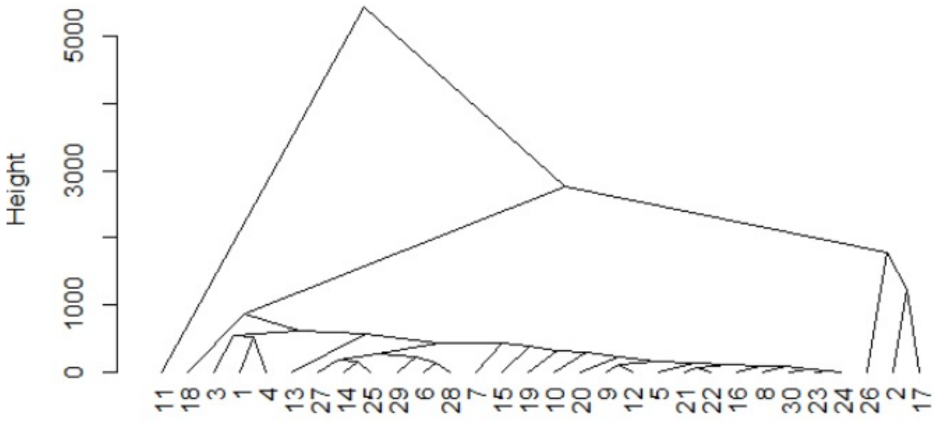
```
#Horizontal Dendrogram
plot(as.dendrogram(bank.nearest_neighbor), labels = NULL, hang = 0.1,
       sub = NULL,
      xlab = "Distance between term deposit values", ylab = NULL,horiz = TRUE,
      main="Horizontal Dendrogram")
```



TRIANGLE DENDROGRAM

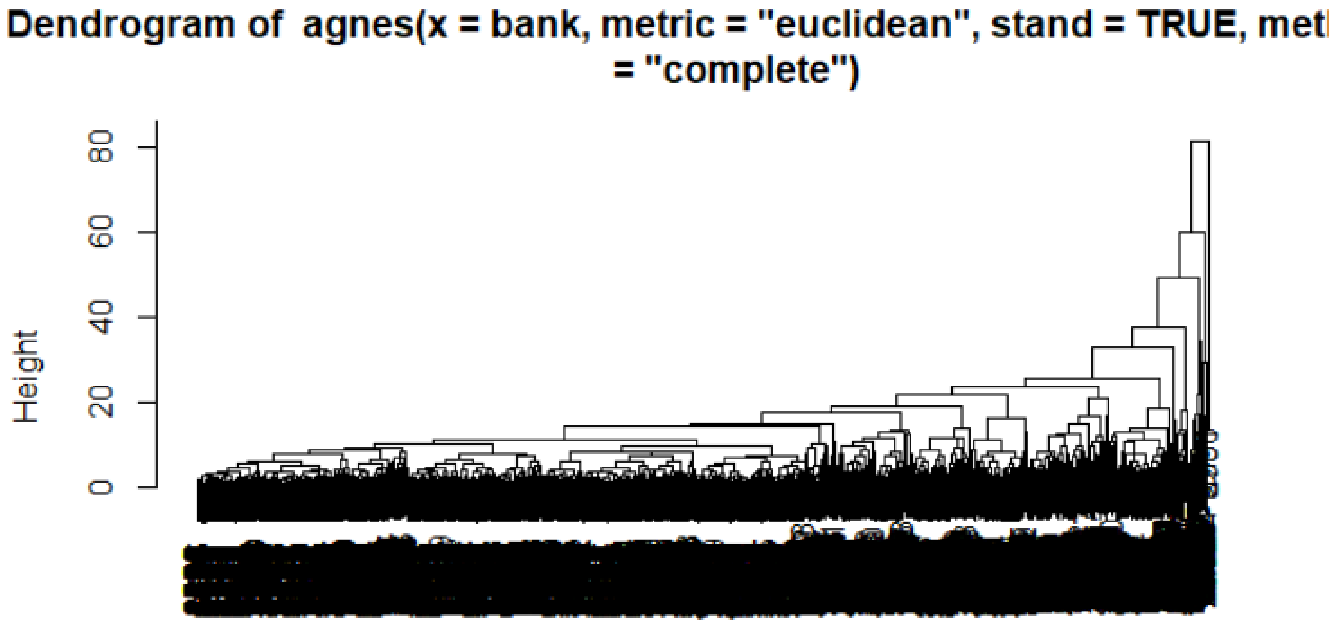


```
#Triangle Plot
plot(as.dendrogram(bank.nearest_neighbor),
     type = "triangle",
     ylab = "Height")
```

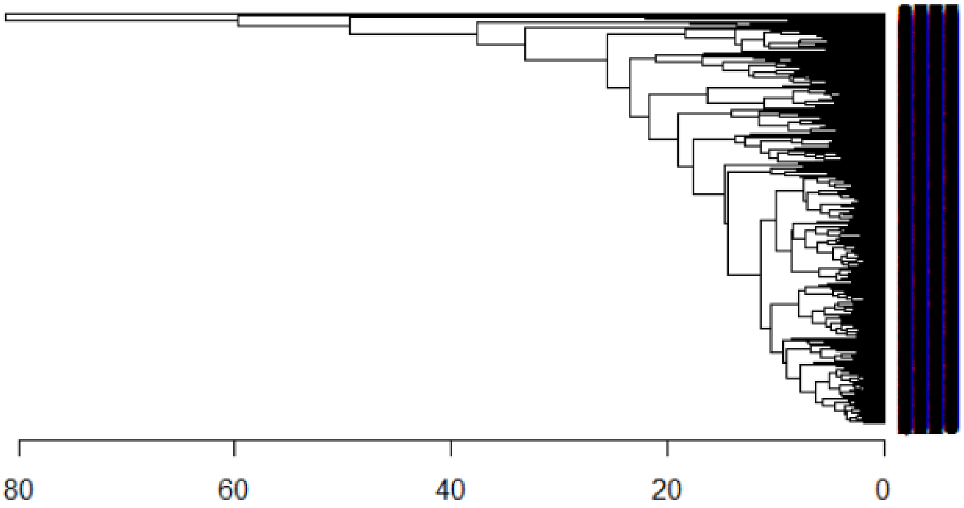


INTERACTIVE PLOTS

First, we applied the agnes function to our original dataset to classified rows. And each row represents a potential client. Here is the result of the classification. We used the Euclidean distance and the complete method for the distance between two clusters.



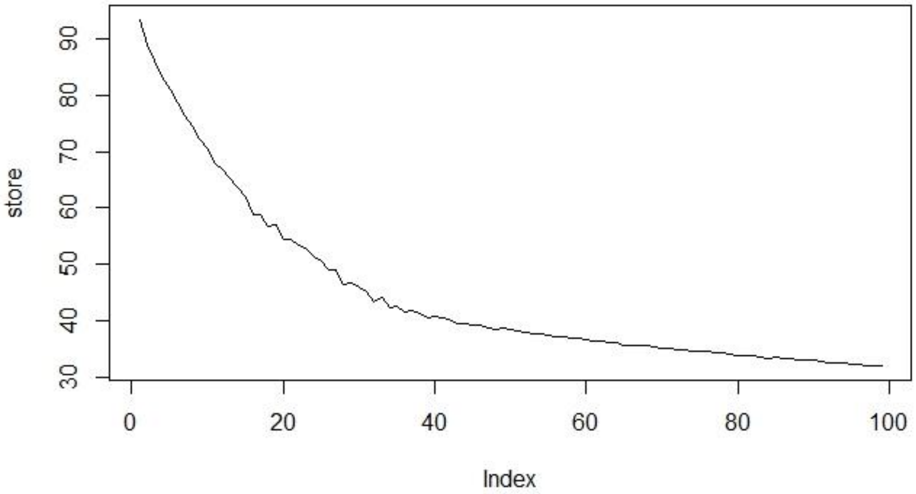
From the plot above, we can see that there are two clusters in early stages. But both have many branches. That means we may not expect outliers in this dataset. Or no client is quite unique compared with others. But this graph is too rough. We then further applied other functions to explore this dataset. Besides, we also get the horizontal dendrogram. Here is the outcome.



K-MEANS CLUSTERING

Before we apply any other functions, we substitute these categorical variables in our dataset with dummy variables. That makes the number of columns increase to 42. And all variables now are numeric. We then scale this expanded dataset with means of each column.

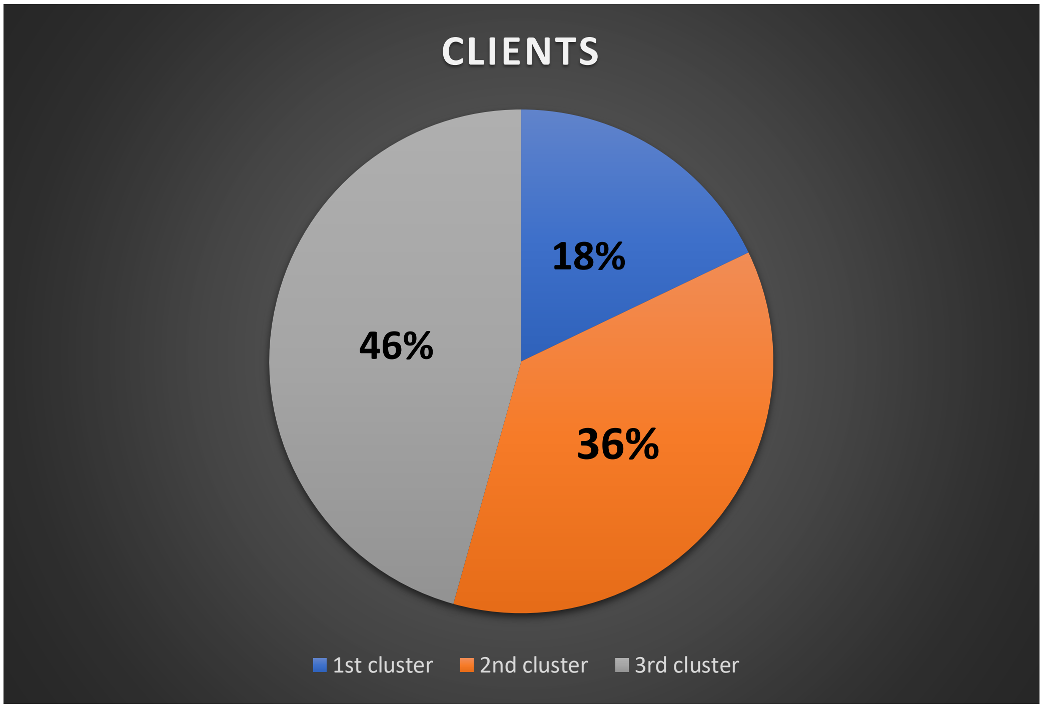
After that, we applied the k-means clustering to the new dataset. The initial random sets are 100. And the maximum of iterations is also 100. We tried 2 to 100 clusters. For each case, we computed the proportion of within-group variation of the total variation to find if there is any threshold. Then we got the following figure.



As you can see, the proportion decreases gradually. And the absolute value of the slope is also decreasing. It means new clusters seem to be similar as more and more clusters are introduced. Based on this figure, we think 40 clusters may be suitable for this dataset. After 40, the absolute value of the slope is obviously smaller. But we have not found an obvious threshold yet.

THREE CLUSTER CASE

We then explore the case with three clusters. We want to know how many customers are in each cluster when there are three clusters. It shows that 89% of the total variation are in groups in this case. There are 809 clients in the first cluster, 1649 for the second one, and 2065 for the third one. Here is the pie chart of clusters. So, the 3rd cluster may need another division since it takes a large part.


CLIENTS
1st cluster
2nd cluster
3rd cluster